

Efficient Semidefinite Spectral Clustering via Lagrange Duality

Yan Yan, Chunhua Shen, and Hanzi Wang, *Senior Member, IEEE*

Abstract—We propose an efficient approach to semidefinite spectral clustering (SSC), which addresses the Frobenius normalization with the positive semidefinite (p.s.d.) constraint for spectral clustering. Compared with the original Frobenius norm approximation based algorithm, the proposed algorithm can more accurately find the closest doubly stochastic approximation to the affinity matrix by considering the p.s.d. constraint. In this paper, SSC is formulated as a semidefinite programming (SDP) problem. In order to solve the high computational complexity of SDP, we present a dual algorithm based on the Lagrange dual formalization. Two versions of the proposed algorithm are proffered: one with less memory usage and the other with faster convergence rate. The proposed algorithm has much lower time complexity than that of the standard interior-point based SDP solvers. Experimental results on both UCI data sets and real-world image data sets demonstrate that 1) compared with the state-of-the-art spectral clustering methods, the proposed algorithm achieves better clustering performance; and 2) our algorithm is much more efficient and can solve larger-scale SSC problems than those standard interior-point SDP solvers.

Index Terms—Spectral clustering, Doubly stochastic normalization, Semidefinite programming, Lagrange duality.

I. INTRODUCTION

CLUSTERING is one of the most popular techniques for statistical data analysis with various applications, including image analysis, pattern recognition, machine learning, and information retrieval [1]. The objective of clustering is to partition a data set into groups (called clusters) such that the data points in the same cluster are more similar than those in other clusters. Numerous clustering algorithms have been developed in the literature [1], such as k -means, single linkage, and fuzzy clustering.

In recent years, spectral clustering [2]–[16], a class of clustering algorithms based on the spectrum analysis of the affinity matrix, has emerged as an effective clustering technique. Compared with the traditional algorithms [1], such as k -means or single linkage, spectral clustering has many fundamental advantages. For example, it is easy to implement and reasonably fast, especially for large sparse matrices [17].

Spectral clustering formulates clustering as a graph partitioning problem without estimating an explicit model of the data distribution. In general, a graph partitioning approach starts with a pairwise affinity matrix, which measures the

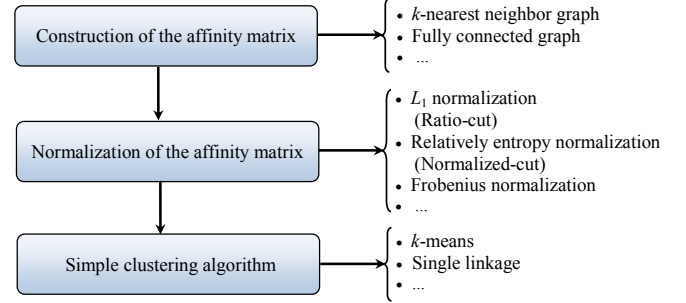


Fig. 1. The three critical factors that affect the final performance of spectral clustering.

degree of similarity between data points, followed by a normalization step. Then, the leading eigenvectors of the normalized affinity matrix are extracted to perform dimensionality reduction for effective clustering in the lower dimensional subspace. Therefore, the three critical factors that affect the final performance of spectral clustering are [18], [19]: 1) the construction of the affinity matrix, 2) the normalization of the affinity matrix, and 3) the simple clustering algorithm, as shown in Fig. 1.

The purpose of the construction of the affinity matrix is to model the neighborhood relationship between data points. There are several popular ways [12] to construct the affinity matrix, such as the k -nearest neighbor graph and the fully connected graph. The normalization of the affinity matrix is achieved by finding the closest doubly stochastic matrix to the affinity matrix under a certain error measure [18]–[20], while the simple clustering algorithm (e.g. k -means) is used to partition an embedded coordinate system (formed by the principal k eigenvectors of the normalized affinity matrix) in an easier and simpler way. Empirical studies [19] indicate that the first two critical factors have a greater impact on the final clustering performance compared with the third critical factor (i.e., the simple clustering algorithm). In this paper, we mainly investigate the second critical factor – the normalization of the affinity matrix for effective spectral clustering.

We briefly review some related work [3]–[7] before presenting our work. Given a similarity graph with the affinity matrix, the simplest way to construct a partition of the graph is to solve the mincut problem [3], which aims to minimize the weights of edges (i.e., the summation of the similarity) between subgraphs. The mincut, however, usually leads to unsatisfactory clustering results due to an inexplicit limit for the size of the subgraph. To circumvent this problem,

Y. Yan is with the School of Information Science and Technology, Xiamen University, Xiamen, 361005, China. (e-mail: yanyan@xmu.edu.cn)

C. Shen is with the Australian Center for Visual Technologies, and School of Computer Science at The University of Adelaide, SA 5005, Australia. (e-mail: chunhua.shen@adelaide.edu.au)

H. Wang is with the School of Information Science and Technology, Xiamen University, Xiamen, 361005, China. (e-mail: hanzhi.wang@xmu.edu.cn)

Ratio-cut [4], [5] and Normalized-cut [7] are the two most common algorithms. In Ratio-cut [4], [5], the size of the subgraph is measured by the number of vertices, whereas, the size is measured by the weight of the edges attached to a subgraph in Normalized-cut. In essence, what Normalized-cut and Ratio-cut try to achieve is to balance the cuts between clusters. Unfortunately, the optimal solution to the above graph partitioning problems is NP hard. An effective approach is to consider the continuous relaxation versions of these problems [2], [7]–[10]. Minmax-cut was proposed in [8] and showed more balanced partitions than Normalized-cut and Ratio-cut. Nie et al. [13] applied an additional nonnegative constraint into Minmax-cut to obtain more accurate clustering results. Recently, a spectral embedding clustering framework [14] was developed to incorporate the linear property of the cluster assignment matrix.

In [18], [19], it has been shown that the key difference between Ratio-cut and Normalized-cut is the error measure used to find the closest doubly stochastic approximation of the input affinity matrix during the normalization step. When repeated, the Normalized-cut process converges to the global optimal solution under the relative entropy measure (also called the Kullback-Leibler divergence), while the L_1 normalization leads to the Ratio-cut clustering. Zass et al. [19] developed a scheme for finding the optimal doubly stochastic matrix under the Frobenius norm. Experimental results [19] have demonstrated that the Frobenius normalization based spectral clustering achieves better performance on various standard data sets than the traditional normalization based algorithms, such as the L_1 normalization and the relative entropy normalization based spectral clustering methods.

The main problem with the Frobenius normalization is that the positive semidefinite (p.s.d.) constraint is neglected during the normalization step, which makes the approximation to the affinity matrix less accurate. On the other hand, the Frobenius normalization with the p.s.d. constraint leads to a semidefinite programming (SDP) problem. Standard interior points based SDP solvers, however, have a complexity of approximately $O(n^{6.5})$ for problems involving a matrix variable of size $n \times n$ and $O(n)$ linear constraints. In the case of clustering, n is the number of data points and we can only solve the problem with a limited number of samples (a few hundred in our experiments). In other words, the complexity of the SDP solver limits the applications of the Frobenius normalization with the p.s.d. constraint. Therefore, in this paper, we focus on developing efficient algorithms to solve the Frobenius normalization with the p.s.d. constraint for spectral clustering, termed Semidefinite Spectral Clustering (SSC) hereafter.

In this paper, we present an efficient and effective algorithm, called LD-SSC, which exploits the Lagrange dual structure to solve the SSC problem. The proposed algorithm seeks a matrix that satisfies both the doubly stochastic and positive semidefinite constraints as closely as possible to the affinity matrix. The formulated optimization refers to the SDP problem which aims to optimize a convex function over the convex cone of symmetric and positive semidefinite matrices [21]. Therefore, the global optimal solution can be approached in the polynomial time. What is more, by exploring the

Lagrange dual form, we are able to apply off-the-shelf eigen-decomposition and gradient descent methods (e.g., L-BFGS-B [22]) to solve the SSC problem in a simple manner. Two versions of the proposed algorithm are given in this paper: one with less memory usage (LD-SSC1) and the other with faster convergence rate (LD-SSC2).

One of the main advantages of our algorithm is that, with the proposed formulation, we can solve the SSC problem in the Lagrange dual form very efficiently. Because the strong duality holds, we can recover the primal variable (i.e., the normalized affinity matrix) from the dual solution. Moreover, the computational complexity of the proposed algorithm is $O(t \cdot n^3)$, which is much lower than the traditional SDP solver $O(n^{6.5})$. Here, t is the number of iterations for convergence. Typically t is around 300 (LD-SSC1) or $t \approx 10 \sim 20$ (LD-SSC2) in our experiments. In summary, the main contributions of our work are summarized as follows:

- 1) The proposed LD-SSC algorithm introduces the p.s.d. constraint into the normalization of the affinity matrix. Semidefinite spectral clustering finds a doubly stochastic and p.s.d. matrix that is the closest to the affinity matrix under the Frobenius norm. Improved clustering accuracy is achieved on various standard data sets.
- 2) We propose an efficient algorithm to semidefinite spectral clustering via Lagrange duality. Our algorithm is much more scalable than the standard SDP solvers. The importance of this development is that it allows us to apply semidefinite spectral clustering to large scale data clustering. Compared with the traditional SDP solvers, our proposed algorithm is significantly more efficient, especially when the number of data points is large.

The rest of the paper is organized as follows: Section II introduces the normalization of the affinity matrix. In Section III, the details of the proposed LD-SSC algorithm are presented. The performance of our algorithm is demonstrated and compared with other state-of-the-art algorithms in Section IV. Finally, we conclude our work in Section V.

II. NORMALIZATION OF THE AFFINITY MATRIX

In this section, we briefly introduce the connection between kernel k -means and spectral clustering [18], [19], [23] and reveal the role of the normalization of the affinity matrix for spectral clustering. We begin by introducing the notation used in this paper.

A matrix is denoted by a bold upper-case letter (\mathbf{X}) and a column vector is denoted by a bold lower-case letter (\mathbf{x}). The set of real $M \times N$ matrices is denoted as $\mathbb{R}^{M \times N}$. Let us denote the space of real matrices as \mathbb{S} . Similarly, we denote the space of $M \times M$ symmetric matrices by \mathbb{S}^M and positive semidefinite matrices by \mathbb{S}_+^M . For a matrix $\mathbf{X} \in \mathbb{S}_+^M$, the following statements are equivalent: (1) $\mathbf{X} \succeq 0$ ($\mathbf{X} \in \mathbb{S}_+^M$); (2) All the eigenvalues of \mathbf{X} are non-negative ($\lambda_i(\mathbf{X}) \geq 0$, $i = 1, \dots, M$); and (3) $\forall \boldsymbol{\mu} \in \mathbb{R}^M$, $\boldsymbol{\mu}^\top \mathbf{X} \boldsymbol{\mu} \geq 0$. The inner product defined on the \mathbb{S}^M is $\langle \mathbf{A}, \mathbf{B} \rangle = \text{Tr}(\mathbf{A}^\top \mathbf{B})$. Here, $\text{Tr}(\cdot)$ is the trace of a symmetric matrix. $\|\cdot\|_F$ denotes the Frobenius

norm, which is defined as

$$\|\mathbf{X}\|_F^2 = \text{Tr}(\mathbf{X}^\top \mathbf{X}) = \sum_{i,j=1}^M x_{i,j}^2.$$

Given a symmetric matrix $\mathbf{X} \in \mathbb{S}^M$, its eigen-decomposition is $\mathbf{X} = \mathbf{U}\Sigma\mathbf{U}^\top$. Here, $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_M]$ is an orthonormal matrix, and $\Sigma = \text{diag}(\lambda_1, \dots, \lambda_M)$ is a diagonal matrix whose entries are the eigenvalues of \mathbf{U} . We can explicitly express the positive part of \mathbf{X} as:

$$\mathbf{X}_+ = \sum_{\lambda_i > 0} (\lambda_i \mathbf{u}_i \mathbf{u}_i^\top),$$

and the negative part of \mathbf{X} as:

$$\mathbf{X}_- = \sum_{\lambda_i < 0} (\lambda_i \mathbf{u}_i \mathbf{u}_i^\top).$$

Clearly, $\mathbf{X} = \mathbf{X}_+ + \mathbf{X}_-$ holds.

Given a set of points $\{(\mathbf{a}_1, \dots, \mathbf{a}_n) | \mathbf{a}_i \in \mathbb{R}^M, i = 1, \dots, n\}$, we attempt to partition the n observations into k sets $\{C_1, \dots, C_k\}$ with n_r points in C_r ($r = 1, \dots, k$). Let $K_{i,j} = \kappa(\mathbf{a}_i, \mathbf{a}_j)$ be a symmetric positive semidefinite affinity function. Here the affinity function transforms the pairwise similarity or the pairwise distance into a graph. Thus, the clustering problem is converted to find a partition based on the affinity matrix as $\mathbf{K} = \{K_{i,j}\}$.

k -means is a standard clustering algorithm that partitions a data set into k clusters. However, a major disadvantage of the k -means algorithm is that it can only find linearly-separable clusters in the input space [23]. To overcome this disadvantage, the kernel k -means algorithm uses a function $\phi(\mathbf{x})$ to map the input vector into a possibly higher-dimensional feature space so that the clusters are linearly separable in the new space. The kernel k -means algorithm seeks to find the clusters so as to minimize the following objective function:

$$\sum_{r=1}^k \sum_{\mathbf{a}_i \in C_r} \|\phi(\mathbf{a}_i) - \mathbf{m}_r\|^2, \quad (1)$$

where the function $\phi(\mathbf{a}_i)$ maps the input vector \mathbf{a}_i into a higher-dimensional feature space and $\mathbf{m}_r = (1/|C_r|) \sum_{\mathbf{a}_i \in C_r} \phi(\mathbf{a}_i)$ is the center of the r -th cluster with $|C_r| = n_r$. After some algebraic manipulations, we can derive that minimizing (1) is equivalent to solve the following problem:

$$\max_{C_1, \dots, C_k} \sum_{r=1}^k \frac{1}{n_r} \sum_{(\mathbf{a}_i, \mathbf{a}_j) \in C_r} \kappa(\mathbf{a}_i, \mathbf{a}_j), \quad (2)$$

where $\kappa(\mathbf{a}_i, \mathbf{a}_j) = \phi(\mathbf{a}_i)^\top \phi(\mathbf{a}_j)$.

Since $K_{i,j} = \kappa(\mathbf{a}_i, \mathbf{a}_j)$, (2) can be converted into the following matrix form [18]:

$$\begin{aligned} \max_{\mathbf{W}} \quad & \text{Tr}(\mathbf{W}^\top \mathbf{K} \mathbf{W}) \\ \text{s.t.} \quad & \mathbf{W} \geq 0, \mathbf{W} \mathbf{W}^\top \mathbf{1} = \mathbf{1}, \mathbf{W}^\top \mathbf{W} = \mathbf{I}, \end{aligned} \quad (3)$$

where $\mathbf{W} \in \mathbb{R}^{n \times k}$ is the desired assignment matrix with $w_{i,j} = 1/\sqrt{n_j}$ if $i \in C_j$, and $w_{i,j} = 0$ otherwise. $\mathbf{1}$ is a column vector, all of whose components are ones. \mathbf{I} is an

identity matrix, whose dimension is clear from the context. Hence, if we obtain a matrix \mathbf{W} that maximizes $\text{Tr}(\mathbf{W}^\top \mathbf{K} \mathbf{W})$ under the above constraints, we can find the solution to kernel k -means.

On the other hand, spectral clustering defines the two-stage approach to the above problem (3). First, the normalized matrix $\hat{\mathbf{K}}$ of the input affinity matrix \mathbf{K} is computed. Then, the spectral decomposition is used to find the solution as follows:

$$\begin{aligned} \max_{\mathbf{W}} \quad & \text{Tr}(\mathbf{W}^\top \hat{\mathbf{K}} \mathbf{W}) \\ \text{s.t.} \quad & \mathbf{W}^\top \mathbf{W} = \mathbf{I}, \end{aligned} \quad (4)$$

whose optimal solution is composed by the principal k eigenvectors of $\hat{\mathbf{K}}$. Typically, the eigenvectors form a new coordinate system in a k -dimensional subspace where the popular clustering approaches, such as k -means, are readily applicable. We refer to the process of transforming \mathbf{K} to the normalized matrix $\hat{\mathbf{K}}$ as the normalization step.

Next, we show that the normalization step in the spectral clustering algorithms, such as Normalized-cut and Ratio-cut, is to find a doubly stochastic matrix as closely as possible to the input affinity matrix under different error measures.

Let $\mathbf{F} \in \mathbb{S}^n$ be a square matrix with $f_{i,j} = 1/n_r$ if $(\mathbf{a}_i, \mathbf{a}_j) \in C_r$, and $f_{i,j} = 0$ otherwise. Here, if we arrange the data points according to the cluster membership, then \mathbf{F} is a block diagonal matrix with the diagonal blocks $\mathbf{F}_1, \dots, \mathbf{F}_k$, where $\mathbf{F}_r = (1/n_r) \mathbf{1} \mathbf{1}^\top$. Obviously, we have $\mathbf{F} = \mathbf{W} \mathbf{W}^\top$. Based on (3), \mathbf{F} satisfies the following constraints [24]:

$$\mathbf{F} \geq 0, \mathbf{F} \mathbf{1} = \mathbf{1}, \mathbf{F} = \mathbf{F}^\top. \quad (5)$$

Here \mathbf{F} is called as the doubly stochastic matrix, which is a square matrix of nonnegative real numbers and the elements in whose rows and columns add up to 1.

The normalization with the form $\hat{\mathbf{K}} = \mathbf{D}^{-1/2} \mathbf{K} \mathbf{D}^{-1/2}$ where $\mathbf{D} = \text{diag}(\mathbf{K} \mathbf{1})$ is used by Normalized-cut. In [18], it has been proved that for any non-negative symmetric matrix $\mathbf{K}^{(0)}$, the iterative process $\mathbf{K}^{(t+1)} = \mathbf{D}^{-1/2} \mathbf{K}^{(t)} \mathbf{D}^{-1/2}$ with $\mathbf{D} = \text{diag}(\mathbf{K}^{(t)} \mathbf{1})$ converges to a doubly stochastic matrix under the relative entropy measure (using the symmetric version of the iterative proportional fitting procedure [24]). Alternatively, the closest doubly stochastic matrix under the L_1 norm is $\hat{\mathbf{K}} = \mathbf{K} - \mathbf{D} + \mathbf{I}$ which leads to Ratio-cut. Zass et al. [19] have shown that it is more natural to find the doubly stochastic matrix under the Frobenius error norm. The Frobenius normalization can be formulated as the following quadratic linear programming (QLP) problem:

$$\begin{aligned} \hat{\mathbf{K}} = \arg \min_{\mathbf{F}} \quad & \|\mathbf{K} - \mathbf{F}\|_F^2 \\ \text{s.t.} \quad & \mathbf{F} \geq 0, \mathbf{F} \mathbf{1} = \mathbf{1}, \mathbf{F} = \mathbf{F}^\top. \end{aligned} \quad (6)$$

In conclusion, kernel k -means and spectral clustering are closely connected where the normalization of the affinity matrix is related to the doubly stochastic constraint induced by kernel k -means.

III. SEMIDEFINITE SPECTRAL CLUSTERING

In this section, we present an efficient algorithm with two versions, which aims at the effective normalization of the affinity matrix for semidefinite spectral clustering.

A. Frobenius Normalization with the P.S.D. Constraint

Empirical studies [18], [19] have shown that the normalization of the affinity matrix \mathbf{K} has significant effects on the final clustering results. Compared with the L_1 normalization and the relative entropy normalization, the Frobenius normalization has been proved to be very practical and can significantly boost the clustering performance. In fact, it is natural to find a doubly stochastic approximation that satisfies the constraints (5) to \mathbf{K} under the Frobenius norm, which is the extension of the common Euclidean vector norm $\|\cdot\|_2$ for the matrix.

A simple derivation yields that \mathbf{F} is a p.s.d. matrix (i.e., $\mathbf{F} \succcurlyeq 0$) since $\mathbf{F} = \mathbf{W}\mathbf{W}^\top$ and $\forall \boldsymbol{\mu} \in \mathbb{R}^n$, $\boldsymbol{\mu}^\top \mathbf{F} \boldsymbol{\mu} = \boldsymbol{\mu}^\top \mathbf{W}\mathbf{W}^\top \boldsymbol{\mu} \geq 0$. However, the p.s.d. constraint is neglected during the normalization step in [19] due to the simplification of the computational complexity. Taking the p.s.d. constraint into consideration will make the doubly stochastic approximation to the affinity matrix more accurate. In other words, the doubly stochastic approximation should satisfy the p.s.d. constraint. Therefore, it is desirable to find a doubly stochastic and p.s.d. matrix that approximates the affinity matrix as closely as possible under the error measure.

The proposed algorithm aims to seek a doubly stochastic and p.s.d. matrix under the Frobenius norm. The optimization problem can be written as follows:

$$\begin{aligned} \hat{\mathbf{K}} = \arg \min_{\mathbf{F}} & \|\mathbf{K} - \mathbf{F}\|_F^2 \\ \text{s.t. } & \mathbf{F} \geq 0, \mathbf{F}\mathbf{1} = \mathbf{1}, \mathbf{F} = \mathbf{F}^\top, \mathbf{F} \succcurlyeq 0, \end{aligned} \quad (7)$$

where the first three constraints make the optimal solution be doubly stochastic while the last constraint forces the final matrix to be p.s.d..

The optimization problem (7) can be converted into an instance of semidefinite programming (SDP) where the matrix is required to be p.s.d., and then be solved by the standard solver packages directly. However, as we discussed earlier, general purpose SDP solvers [21] are computationally expensive and only small scale problems is applicable in a reasonable time. Thus, it is necessary to design an alternative algorithm that can greatly reduce the computational complexity while at the same time, achieving comparable performance. In the next subsection, an efficient algorithm to solve the above problem by exploiting the Lagrange dual problem of (7) is presented.

B. Semidefinite Spectral Clustering via Lagrange Duality

The Lagrange duality takes the constraints in the primal form into consideration by augmenting the objective function with a weighted sum of the constraint functions. To derive the Lagrange dual of (7), we introduce the symmetric matrix \mathbf{Z} and \mathbf{Q} to associate with the p.s.d. constraint $\mathbf{F} \succcurlyeq 0$ and the non-negative constraint $\mathbf{F} \geq 0$, respectively. The two variables \mathbf{u}_1 and \mathbf{u}_2 associate with the equality constraints in the primal form. The Lagrangian of (7) is then written as:

$$\begin{aligned} \underbrace{\ell(\mathbf{F})}_{\text{primal}}, \underbrace{\mathbf{Z}, \mathbf{Q}, \mathbf{u}_1, \mathbf{u}_2}_{\text{dual}} &= \frac{1}{2} \|\mathbf{K} - \mathbf{F}\|_F^2 - \langle \mathbf{F}, \mathbf{Q} \rangle - (\mathbf{F}\mathbf{1} - \mathbf{1})^\top \mathbf{u}_1 \\ &\quad - (\mathbf{F}^\top \mathbf{1} - \mathbf{1})^\top \mathbf{u}_2 - \langle \mathbf{F}, \mathbf{Z} \rangle, \end{aligned} \quad (8)$$

with $\mathbf{Z} \succcurlyeq 0$ and $\mathbf{Q} \geq 0$.

Because $\mathbf{F} = \mathbf{F}^\top$, we have $\mathbf{u}_1 = \mathbf{u}_2 = \mathbf{u}$. Based on the Karush-Kuhn-Tucker (KKT) optimality conditions [21], we minimize the Lagrangian over \mathbf{F} which means that its gradient is set to zero and then we have

$$\frac{\partial \ell(\mathbf{F}, \mathbf{Z}, \mathbf{Q}, \mathbf{u})}{\partial \mathbf{F}} = \mathbf{F} - \mathbf{K} - \mathbf{Q} - \mathbf{u}\mathbf{1}^\top - \mathbf{1}\mathbf{u}^\top - \mathbf{Z} = \mathbf{0}. \quad (9)$$

Therefore, the connection between the primal and dual variables is given by

$$\mathbf{F}^* = \mathbf{K} + \mathbf{Q}^* + \mathbf{u}^* \mathbf{1}^\top + \mathbf{1} \mathbf{u}^{*\top} + \mathbf{Z}^*. \quad (10)$$

Based on the above expression for \mathbf{F} , the dual function is

$$\begin{aligned} g(\mathbf{Z}, \mathbf{Q}, \mathbf{u}) &= \inf_{\mathbf{F}} \ell(\mathbf{F}, \mathbf{Z}, \mathbf{Q}, \mathbf{u}) \\ &= \frac{1}{2} \|\mathbf{K} - \mathbf{F}\|_F^2 - \langle \mathbf{F}, \mathbf{Q} \rangle - \langle \mathbf{F}, \mathbf{Z} \rangle \\ &= \frac{1}{2} \|\mathbf{Z} + \mathbf{Q} + \mathbf{M}\|_F^2 - \langle \mathbf{F}, \mathbf{Z} + \mathbf{Q} + \mathbf{M} \rangle + 2\mathbf{1}^\top \mathbf{u} \\ &= -\frac{1}{2} \|\mathbf{Z} + \mathbf{Q} + \mathbf{M} + \mathbf{K}\|_F^2 + \frac{1}{2} \|\mathbf{K}\|_F^2 + 2\mathbf{1}^\top \mathbf{u}, \end{aligned} \quad (11)$$

where $\mathbf{M} = \mathbf{u}\mathbf{1}^\top + \mathbf{1}\mathbf{u}^\top$. The above equation is derived by using the fact that $\langle \mathbf{F}, \mathbf{u}\mathbf{1}^\top \rangle = \text{Tr}(\mathbf{F}\mathbf{1}\mathbf{u}^\top) = \mathbf{u}^\top \mathbf{1}$ and $\langle \mathbf{F}, \mathbf{1}\mathbf{u}^\top \rangle = \text{Tr}(\mathbf{F}\mathbf{1}\mathbf{u}^\top) = \mathbf{u}^\top \mathbf{1}$.

So, the dual formulation becomes

$$\begin{aligned} \max_{\mathbf{Z}, \mathbf{Q}, \mathbf{u}} & -\frac{1}{2} \|\mathbf{Z} + \mathbf{Q} + \mathbf{M} + \mathbf{K}\|_F^2 + \frac{1}{2} \|\mathbf{K}\|_F^2 + 2\mathbf{1}^\top \mathbf{u} \\ \text{s.t. } & \mathbf{Z} \succcurlyeq 0, \mathbf{Q} \geq 0. \end{aligned} \quad (12)$$

Both the primal and Lagrange dual problems are convex. Under mild conditions, the Slater's condition holds, which means the strong duality between the primal and dual problems. It also implies that the duality gap is zero. As a result, we are able to indirectly solve the primal by solving the dual problem. In addition, the KKT optimality conditions (which are necessary and sufficient conditions for any pair of primal and dual optimal points of a convex problem) enable us to recover the primal variable from the dual solution in our case, as shown in (10).

Since $\|\mathbf{K}\|_F^2$ is a constant, (12) can be further simplified as

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{Q}, \mathbf{u}} & \frac{1}{2} \|\mathbf{Z} + \mathbf{Q} + \mathbf{M} + \mathbf{K}\|_F^2 - 2\mathbf{1}^\top \mathbf{u} \\ \text{s.t. } & \mathbf{Z} \succcurlyeq 0, \mathbf{Q} \geq 0. \end{aligned} \quad (13)$$

Problem (13) still has the p.s.d. constraint and it is not immediately clear about how to solve the problem efficiently other than using off-the-shelf SDP solvers. One solution is coordinate ascent. By taking the idea of the cyclic coordinate ascent technique [25] (which seeks for the optimum of the objective function by repeatedly optimizing each of the coordinate directions), we can efficiently solve (13).

In particular, if we fix \mathbf{Q} and \mathbf{u} , the dual problem (13) becomes

$$\begin{aligned} \min_{\mathbf{Z}} & \frac{1}{2} \|\mathbf{Z} + \mathbf{Q} + \mathbf{M} + \mathbf{K}\|_F^2 \\ \text{s.t. } & \mathbf{Z} \succcurlyeq 0. \end{aligned} \quad (14)$$

If we define a symbol $\mathbf{P} = -(\mathbf{Q} + \mathbf{M} + \mathbf{K})$ which is a function of \mathbf{Q} and \mathbf{u} , then (14) is to minimize $\|\mathbf{Z} - \mathbf{P}\|_F^2$

such that \mathbf{Z} satisfies the p.s.d. constraint. It has a closed-form solution $\mathbf{Z}^* = \mathbf{P}_+$, where \mathbf{P}_+ is the positive part of \mathbf{P} .

Hence, (13) is simplified into

$$\begin{aligned} \min_{\mathbf{Q}, \mathbf{u}} \quad & \frac{1}{2} \|\mathbf{P}_- \|_F^2 - 2\mathbf{1}^\top \mathbf{u} \\ \text{s.t.} \quad & \mathbf{Q} \geq 0, \end{aligned} \quad (15)$$

where $\mathbf{P}_- = \mathbf{P} - \mathbf{P}_+$.

Given a fixed \mathbf{Q} , we can write the above optimization problem as follows:

$$\min_{\mathbf{u}} \quad \frac{1}{2} \|\mathbf{P}_- \|_F^2 - 2\mathbf{1}^\top \mathbf{u}, \quad (16)$$

where the objective function can be proved to be differentiable (see [26] for details). So, (16) can be easily solved by using a gradient descent method (e.g. L-BFGS-B [22]) since it does not have the matrix variables. L-BFGS-B is a limited-memory quasi-Newton algorithm for solving bound-constrained nonlinear optimization problems.

On the other hand, given fixed \mathbf{u} and \mathbf{Z} , (13) becomes

$$\begin{aligned} \min_{\mathbf{Q}} \quad & \frac{1}{2} \|\mathbf{Q} + \mathbf{Z} + \mathbf{M} + \mathbf{K}\|_F^2 \\ \text{s.t.} \quad & \mathbf{Q} \geq 0. \end{aligned} \quad (17)$$

Problem (17) has a closed-form solution which is $\mathbf{Q}^* = \text{thr}_{\geq 0}(-(\mathbf{Z} + \mathbf{M} + \mathbf{K}))$. Here, $\text{thr}_{\geq 0}(\mathbf{X})$ is an operator that zeros out all the negative entries of \mathbf{X} .

To use L-BFGS-B, we need to implement the callback function of L-BFGS-B, which computes the gradient of the objective function of (16). The gradient of the dual problem (16) can be calculated as:

$$g(u_i) = -2 - \langle \mathbf{P}_-, \hat{\mathbf{T}}_i \rangle, i = 1, \dots, n. \quad (18)$$

Here $\hat{\mathbf{T}}_i = \mathbf{T}_i + \mathbf{T}_i^\top$, where \mathbf{T}_i is an $n \times n$ zero matrix except that all the elements in the i -th row are ones.

In summary, problem (13) can be solved by alternatively optimizing \mathbf{Z} , \mathbf{Q} and \mathbf{u} , where in each iteration one variable is optimized while fixing all other variables. One version of the proposed algorithm is given in Algorithm 1 (denoted as LD-SSC1).

In (18), to obtain the gradient for each variable u_i ($i = 1, \dots, n$), we need to compute the inner product between \mathbf{P}_- and $\hat{\mathbf{T}}_i$ at each iteration. Here, \mathbf{P}_- is a function of the variable \mathbf{u} . Note that, the computation of \mathbf{P}_- involves the eigen-decomposition and \mathbf{P}_- needs to be calculated to evaluate all the gradients at each iteration of L-BFGS-B. When the number of constraints is not far more than the number of data points, the eigen-decomposition dominates the computational complexity at each iteration. Therefore, the overall complexity of LD-SSC1 is $O(t \cdot n^3)$. Here, t is the number of iterations for convergence (typically $t \approx 250 \sim 1,000$ in our experiments); n is the number of data points. To be specific, the number of iterations for the inner loop (L-BFGS-B [22] is employed in our case in Step 2) is $5 \sim 10$, while the number of iterations for the outer loop (Step 2 to Step 4) is $50 \sim 100$.

The above optimization takes the dual variables into consideration individually, but (15) can also be directly solved by L-BFGS-B with the variables \mathbf{Q} and \mathbf{u} altogether, since we

Algorithm 1 LD-SSC1.

Input: Given a set of points $\{(\mathbf{a}_1, \dots, \mathbf{a}_n) | \mathbf{a}_i \in \mathbb{R}^M, i = 1, \dots, n\}$ and the number k of clusters.

Output: Clusters $\{C_1, \dots, C_k\}$ with n_r points in C_r .

- 1: Construct a similarity graph (e.g., k -nearest neighbor graph or fully connected graph) based on the given set; Initialize $\mathbf{Q} = \mathbf{I}$;
 - 2: Optimize (16) to get \mathbf{u} using L-BFGS-B with the gradient (18) of the objective function;
 - 3: Calculate \mathbf{Z} by using $\mathbf{Z} = \mathbf{P}_+$, and \mathbf{Q} by using $\mathbf{Q} = \text{th}_{\geq 0}(\mathbf{X})$;
 - 4: Go to Step 2 until the algorithm converges;
 - 5: Obtain the final $\hat{\mathbf{K}}$ according to (10);
 - 6: Compute the first k eigenvectors of $\hat{\mathbf{K}}$;
 - 7: Cluster the points in the k -dimensional subspace using the simple clustering algorithm.
-

Algorithm 2 LD-SSC2.

Input: Given a set of points $\{(\mathbf{a}_1, \dots, \mathbf{a}_n) | \mathbf{a}_i \in \mathbb{R}^M, i = 1, \dots, n\}$ and the number k of clusters.

Output: Clusters $\{C_1, \dots, C_k\}$ with n_r points in C_r .

- 1: Construct a similarity graph (e.g. k -nearest neighbor graph or fully connected graph) based on the given set;
 - 2: Optimize (15) to get \mathbf{Q} and \mathbf{u} using L-BFGS-B with the gradients calculated by (18) and (19);
 - 3: Calculate \mathbf{Z}^* by using $\mathbf{Z}^* = \mathbf{P}_+$ based on the outputs (i.e., \mathbf{Q}^* and \mathbf{u}^*) of the optimization step in Step 2;
 - 4: Obtain the final $\hat{\mathbf{K}}$ according to (10);
 - 5: Compute the first k eigenvectors of $\hat{\mathbf{K}}$;
 - 6: Cluster the points in the k -dimensional subspace using the simple clustering algorithm.
-

can easily obtain the gradients of the objective function of (15) over \mathbf{Q} and \mathbf{u} , which are

$$g(\mathbf{Q}) = -\mathbf{P}_- \quad (19)$$

and (18), respectively. Thus, we have another efficient version to solve the SSC problem, which is given in Algorithm 2 (denoted as LD-SSC2).

Compared with LD-SSC1, the main difference between LD-SSC2 and LD-SSC1 is in that LD-SSC2 is more efficient since the outer loop is removed during the optimization. More specifically, when evaluating the gradients over \mathbf{Q} and \mathbf{u} at each iteration, both gradients need to compute the \mathbf{P}_- that involves the eigen-decomposition. Similar to LD-SSC1, the eigen-decomposition dominates the computational complexity of LD-SSC2. Therefore, the overall complexity of LD-SSC2 is $O(t' \cdot n^3)$. Here, t' is the number of iterations for convergence and typically $t' = 10 \sim 20$ in our experiments. Compared with LD-SSC1, the computational complexity of LD-SSC2 is lower. However, because the variables \mathbf{Q} and \mathbf{u} are jointly optimized, LD-SSC2 requires more memory usage than LD-SSC1 during each iteration in the optimization step (cf. (15) and (16)).

C. Discussions

There are a few important issues on the proposed LD-SSC1 and LD-SSC2 algorithms.

- First, LD-SSC and the original Frobenius normalization based spectral clustering [19] are intrinsically different in the formulated optimization problems and hence the solutions are different. On one hand, the optimization problem in [19] is formulated as a quadratic programming problem. In contrast, our formulation is an SDP problem. Compared with the work in [19] that only tries to find a closet doubly stochastic matrix, the proposed LD-SSC emphasizes the importance of the p.s.d. property of the normalization matrix, which makes the approximation to the input affinity matrix more accurate. On the other hand, in [19], the Von-Neumann's successive projections lemma [27] is applied to solve the quadratic problem. Our proposed LD-SSC, however, solves the SDP problem by exploiting the Lagrange dual form.
- Second, Xing et al. [28] proposed the semidefinite relaxation for k Normalized-cut. Our purposed LD-SSC method is different from Xing et al. The SDP relaxation to k Normalized-cut gives only a tighter lower bound on the cut weight compared to the traditional spectral relaxation. In contrast, LD-SSC mainly focuses on the normalization step for solving the SSC problem.
- Third, to deal with the large-scale SDP problem, LD-SSC exploits the duality property. Several methods [29], [30] have also been proposed to solve large-scale SDP problems. For example, in [29], matrix factorization is used to approximate a large-scale SDP problem with a smaller one. Note that in our case, we *exactly* solve the original SDP problem.
- Lastly, Luo et al. [16] developed a graph learning algorithm by solving a convex optimization problem with the low rank and p.s.d. constraints. Our algorithm and Luo et al.'s work present two different approaches to obtain a good normalization. An efficient algorithm based on augmented Lagrangian multiplier was proposed to attain the global optimum in [16], while LD-SSC takes advantages of the Lagrange duality property.

Next, we test the proposed methods on various data sets.

IV. EXPERIMENTS

In order to evaluate the proposed LD-SSC algorithm (two versions: i.e., LD-SSC1 and LD-SSC2), we conduct a set of clustering experiments across the popular data sets. The following subsections describe the details of the experiments and results.

A. Data Sets

We use several well-studied data sets from the UCI machine learning repository¹ (including SPECTF heart, Wine, Pima, Hayes-Roth, Iris, and BUPA), the cancer data sets² (including Leukemia and Lung), two public face data sets (including

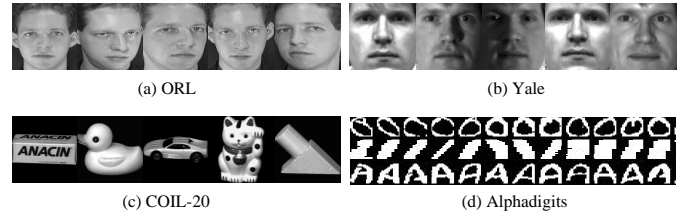


Fig. 2. Sample images from the four image data sets used in the experiments.

ORL face database³ and Yale face database⁴), and two object image data sets (including COIL-20 [32]⁵ and the handwritten binary Alphadigits data set⁶), respectively. The UCI repository is well established and widely used for benchmarking different clustering algorithms. The cancer data sets are the challenging benchmark in the cancer community. The last four data sets are the commonly used real-world image data sets. ORL exhibits the variations in facial expressions and poses while Yale shows various lighting conditions; COIL-20 contains the variations in the viewpoint of objects and the Alphadigits data set exhibits the variations in the shape of handwritten digits and letters. Sample images from the last four image data sets are shown in Fig. 2. Table I summarizes the detailed information and kernel settings for all the data sets.

B. Parameter Settings and Evaluation Metric

In this section, we evaluate the multi-class spectral clustering described in [31] which iteratively solves a discrete solution by using an alternating optimization procedure taking the k principal eigenvectors. Other methods (such as [2]) can also be used, but these methods give similar results [19]. Hence, we employ the framework of [31] while replacing different normalization algorithms in our experiments. Note that the results of all the clustering algorithms depend on the initialization. To reduce statistical variation, we repeat all the clustering algorithms for 10 times with random initialization, and report the results corresponding to the best objective values (similar to [31]).

Two types of kernels used to construct the affinity matrix are the Gaussian kernel and the polynomial kernel. The similarity function for the Gaussian kernel can be written as $K_{i,j} = \exp(-\|\mathbf{a}_i - \mathbf{a}_j\|^2 / \delta^2)$, where the parameter δ controls the width of the neighborhoods. The similarity function for the polynomial kernel is $K_{i,j} = (\mathbf{a}_i^\top \mathbf{a}_j + 1)^d$, where the parameter d represents the degree of the polynomial that affects the shape of the curve. In this paper, in order to achieve the best performance for clustering, the kernel type and kernel parameter are manually chosen for each data set.

For the UCI repository and the cancer data sets, the extracted features are available in the data sets. In contrast, the face data sets, ORL and Yale, are only provided with the raw images. ORL has 40 subjects, where each subject contains 10 images with the size 92×112 . Yale has 11 images for each of

³ORL: <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>

⁴Yale: <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>

⁵COIL-20: <http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php>

⁶Alphadigits: <http://cs.nyu.edu/~roweis/data/binaryalphadigits.mat>

¹UCI Repository: <http://www.ics.uci.edu/~mllearn/MLRepository.html>

²Cancer Data Sets: <http://datam.i2r.a-star.edu.sg/datasets/krbd/>

TABLE I

SUMMARY OF THE DATA SETS AND KERNELS USED IN OUR EXPERIMENTS. THE FIRST SIX DATA SETS ARE ADOPTED FROM THE UCI MACHINE LEARNING REPOSITORY ¹. THE FOLLOWING TWO, LEUKEMIA AND LUNG, ARE FROM THE CANCER DATA SETS ², TWO FACE DATA SETS ARE ORL ³ AND YALE ⁴, AND THE LAST TWO ARE COIL-20 ⁵ AND THE HANDWRITTEN BINARY ALPHADIGITS ⁶ DATA SET.

Data set	#Samples	#Features	#Clusters	Kernel	#Dimension after PCA
SPECTF	267	44	2	Gaussian	—
Wine	178	13	3	Gaussian	—
Pima	768	8	2	Gaussian	—
Hayes-Roth	160	5	3	Gaussian	—
Iris	150	4	3	Gaussian	—
BUPA	345	6	2	Polynomial	—
Leukemia	72	7, 129	2	Polynomial	5
Lung	181	12, 533	2	Polynomial	5
ORL	400	10, 304	40	Polynomial	5
Yale	165	16, 384	15	Polynomial	5
COIL-20	1, 440	1, 024	20	Gaussian	—
Alphadigits	1, 404	320	36	Gaussian	—

TABLE II

THE LOWEST ERROR RATE AND MEAN ERROR RATE OBTAINED BY THE DIFFERENT ALGORITHMS ON THE UCI REPOSITORY. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Algorithm	Lowest Error Rate						Mean Error Rate					
	SPECTF	Wine	Pima	Hayes-Roth	Iris	BUPA	SPECTF	Wine	Pima	Hayes-Roth	Iris	BUPA
NO	0.3408	0.2753	0.3620	0.5250	0.0933	0.4232	0.4173	0.3379	0.4310	0.6078	0.1117	0.4328
RC	0.2322	0.3146	0.4609	0.5313	0.1467	0.4261	0.3278	0.4438	0.4819	0.6031	0.3467	0.4778
NC	0.2472	0.4213	0.3594	0.5500	0.1067	0.4232	0.3728	0.4398	0.4175	0.6119	0.1129	0.4378
FSC	0.2172	0.3427	0.3542	0.5563	0.0933	0.4232	0.2710	0.3903	0.3819	0.5856	0.1108	0.4259
LD-SSC1	0.1873	0.2697	0.3411	0.4688	0.0867	0.4203	0.2231	0.3383	0.3523	0.5463	0.0928	0.4226
LD-SSC2	0.1873	0.2697	0.3411	0.4688	0.0867	0.4203	0.2231	0.3385	0.3523	0.5463	0.0929	0.4228

TABLE III

THE LOWEST ERROR RATE AND MEAN ERROR RATE OBTAINED BY THE DIFFERENT ALGORITHMS ON THE CANCER DATA SETS, THE FACE DATA SETS, AND TWO OBJECT IMAGE DATA SETS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Algorithm	Lowest Error Rate						Mean Error Rate					
	Leukemia	Lung	ORL	Yale	COIL-20	Alphadigits	Leukemia	Lung	ORL	Yale	COIL-20	Alphadigits
NO	0.2917	0.1713	0.8100	0.4242	0.3819	0.6439	0.4506	0.1782	0.8257	0.4533	0.4296	0.6673
RC	0.3333	0.1050	0.8150	0.6909	0.4875	0.7355	0.4475	0.1540	0.8150	0.7261	0.5018	0.7437
NC	0.3750	0.1768	0.9250	0.4848	0.3458	0.6410	0.4275	0.1844	0.9303	0.5158	0.4073	0.6712
FSC	0.3111	0.0994	0.8275	0.4182	0.3583	0.6474	0.3744	0.2735	0.8327	0.4509	0.4116	0.6699
LD-SSC1	0.1806	0.0994	0.6400	0.4182	0.2431	0.4950	0.3148	0.1499	0.6560	0.4479	0.3412	0.5789
LD-SSC2	0.1806	0.0994	0.6450	0.4182	0.2431	0.4932	0.3148	0.1499	0.6560	0.4479	0.3412	0.5789

15 subjects. All the images in Yale are normalized to 128×128 [35]. For simplicity, we extract the feature vector of each face image by lexicographic ordering of the pixel elements of the image. Note that the feature vectors from the cancer data sets and the face data sets are both high-dimensional data, as shown in Table I. To effectively perform spectral clustering, the dimensionality reduction technique is used for preprocessing. In this paper, we use the principal component analysis (PCA) [34] to perform dimensionality reduction. Other sophisticated dimensionality reduction algorithms can also be applied.

The COIL-20 data set [32] has 1,440 images of 20 object categories. Each category contains 72 images. All the images are normalized to 32×32 pixel arrays with 256 gray levels per pixel and then transformed to a 1,024-D feature vector. The binary handwritten Alphadigits data set contains the binary handwritten digits and capital letters. There are 36 classes (including digits of '0' through '9' and capital letters of 'A' through 'Z'). Each class has 39 samples, and each binary image is 20×16 in resolution, which results in a 320-D feature vector.

We implement the proposed algorithm in Matlab and the L-BFGS-B part is in Fortran and Matlab interface. All the

computational time is reported on a desktop with Intel i7 (2.20GHz) CPUs and 4.00 GB RAM.

The clustering performance is evaluated by the error rate. Given that r_i and s_i are the obtained cluster label and the ground truth label, respectively, the error rate is defined as follows [33]:

$$\text{Error Rate} = 1 - \frac{\sum_i^n \delta(s_i, \text{map}(r_i))}{n},$$

where n is the total number of data points; $\delta(a, b)$ is the delta function that equals to one if $a = b$ and zero otherwise. The function $\text{map}(r_i)$ is the permutation mapping function that maps the cluster label r_i to the ground truth label. We choose the lowest error rate [19] and the mean error rate across different kernel parameters (δ in the Gaussian kernel and d in the polynomial kernel) as the evaluation metric in our experiments.

C. Comparisons with State-of-the-art Algorithms

We perform a comparison between the proposed LD-SSC1, LD-SSC2 and spectral clustering with three different state-of-the-art normalization algorithms, including: 1) Normalized-cut (NC) [7], [18] which is based on the relative entropy

normalization; 2) Ratio-cut (RC) [4] which is based on the L_1 normalization; and 3) the Frobenius normalization based spectral clustering (FSC) [19]. To show the importance of the normalization step, we also give the clustering results when no normalization (NO) is applied.

Tables II and III show the comparison results (including the lowest error rate and mean error rate) obtained by the competing algorithms on various data sets. The proposed LD-SSC1 and LD-SSC2 give better or comparable results against the state-of-the-art algorithms like NO, RC, NC, and FSC in terms of the lowest error rate and mean error rate. LD-SSC1 and LD-SSC2 have achieved similar performance since both algorithms optimize the same objective function. It is worth noting that the algorithms, such as NC or RC, can worsen the performance of clustering compared with that without the normalization step for some data sets (e.g. Iris, BUPA, Yale, COIL-20, and Alphadigits). The reason may be that the L_1 norm or the relative entropy measure are not good error measures for the normalization in these data sets.

Fig. 3 shows the clustering performance of different algorithms on all the data sets under varying kernel parameters (δ or d). From the experimental results, we can see that LD-SSC1 and LD-SSC2 consistently outperform the other algorithms for most data sets on the UCI repository. FSC, LD-SSC1, and LD-SSC2 obtain the similar error rate in Iris and BUPA. Some algorithms, such as NO and RC, show great variations in the error rate when the kernel parameter changes, especially for Iris and BUPA. Improved clustering results are achieved by LD-SSC1 and LD-SSC2 for Leukemia and Lung. FSC, however, has achieved a high error rate with some parameters for Lung. LD-SSC1 and LD-SSC2 outperform the other algorithms on the ORL database while FSC, LD-SSC1 and LD-SSC2 achieve similar performance on the Yale database. For the COIL-20 and Alphadigits data sets, both LD-SSC1 and LD-SSC2 obtain the lowest error rate and mean error rate by about 10% ~ 15% less compared with the other algorithms. Again, we observe that the L_1 normalization or the relative entropy normalization degrade the performance of the algorithms when it is compared with that without normalization for some data sets. However, the Frobenius normalization based algorithms (FSC, LD-SSC1, and LD-SSC2) can boost the performance in most data sets. LD-SS1 and LD-SS2 achieve very close performance in most cases.

In Fig. 4, we plot the embedded results (formed by the two principal eigenvectors of the normalized affinity matrix) obtained by the different clustering algorithms for the cancer data sets (i.e., Leukemia and Lung) given the fixed kernel parameter ($d = 2$). The first two principal components (PCs) of the cancer data after the PCA preprocessing step are also given for comparison. The ground truth distribution of the two classes are plotted with different colors. Ideally, after the normalization step, the points in the low-dimensional subspace with higher similarity form closer cluster and the points with lower similarity are far apart from each other. Thus, the data points can be partitioned in an easier and simpler way. For example, most algorithms show improved data distribution results after the normalization step for the

lung data set. However, we find that RC (for Leukemia and Lung) or FSC (for Leukemia) have some outliers after normalization, which makes the following clustering more difficult. NC mixes the two classes together after the normalization step for the Leukemia data set. LD-SSC1 and LD-SSC2 have better clustering results for the Leukemia data set, while the performance of these two methods for the Lung data set seems to be not very separable. This is due to the use of only the first two PCs. However, from Fig. 4, it shows that LD-SSC1 and LD-SSC2 try to align all the points in a line so that the two classes are well separated, which is similar to the idea of the linear discriminant analysis (LDA) [35] for the two-classes case in the supervised learning.

Clustering results show that in most cases, the Frobenius normalization with the p.s.d. constraint (i.e., LD-SSC1 and LD-SSC2) achieves better results than that without the p.s.d. constraint (i.e. FSC) on various data sets. To further demonstrate that the p.s.d. constraint is necessary for more accurate doubly stochastic approximation, we show the comparison of the obtained normalized affinity matrix between FSC and LD-SSC1 on COIL-20 and Alphadigits (given the fixed kernel parameter) in Fig. 5. For convenience, we only show four classes from the two data sets. The original affinity matrix has four dense clusters with overlaps between them. However, after the normalization step, the overlap between clusters are greatly reduced (see Fig. 5(c) and Fig. 5(f)). The connections between different clusters are suppressed while the connections within the same clusters are enhanced, which has a similar effect as the aggregation network [36]. LD-SSC1 (LD-SSC2) obtains a better normalized affinity matrix than FSC. Therefore, by taking the p.s.d. constraint into account, the doubly stochastic approximation to the affinity matrix is more accurate for the Frobenius normalization, which results in better clustering performance.

In summary, experimental results on various data sets show that a good error measure during the normalization can influence the final clustering results. The Frobenius norm is more natural than other error measures, and the proposed LD-SSC1 and LD-SSC2 achieve better clustering results compared to the state-of-the-art algorithms in most cases. The error rate varies with different kernel parameters which controls the intrinsic affinity relationships between data points. In short, LD-SSC1 and LD-SSC2 are more stable than the other competing algorithms in terms of the error rate when the kernel parameter (δ in the Gaussian kernel or d in the polynomial kernel) changes.

D. Computational Complexity

The optimization problem of SSC is a semidefinite programming (SDP) problem, which allows us to use off-the-shelf SDP solvers. To show the efficiency of the proposed algorithm, we also compare the computational time between our scalable LD-SSC algorithm (LD-SSC1 and LD-SSC2) and the SSC using CVX (CVX-SSC) [37], which is a standard package for convex optimization. There are two solvers provided in CVX - SeDumi and SDPT3. We find that the SDPT3 solver is faster than the SeDumi solver for the SDP problem. Therefore, we use CVX with the SDPT3 solver for comparison. Note

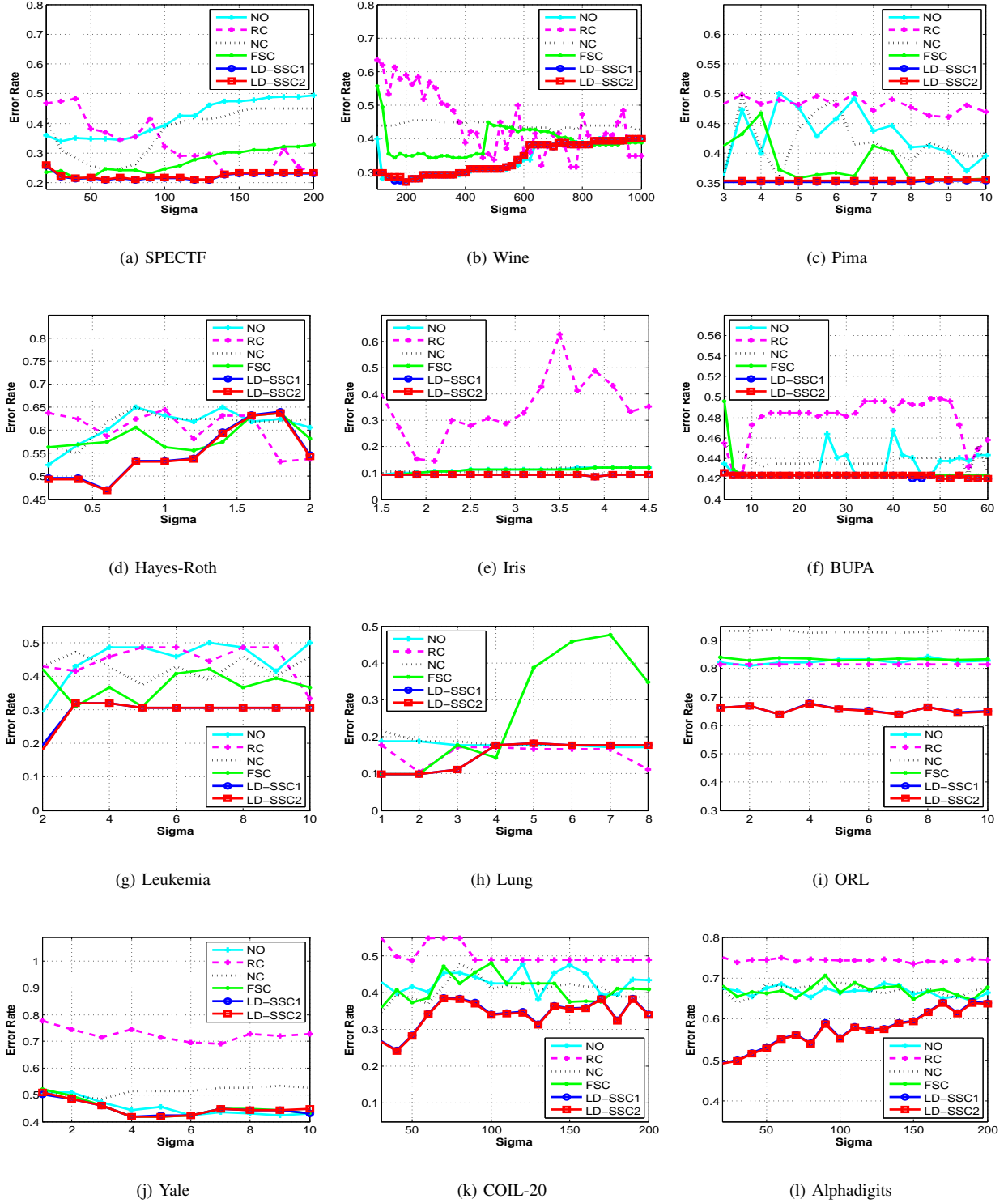


Fig. 3. Error rate vs. similarity measure, for all the data sets used in the experiments.

that, the running time of FSC is not reported. In general, it is much faster than our methods. Because our methods solve much more complicated optimization problems due to the introduction of the p.s.d. constraint.

First, we show the obtained results on synthetic toy data. We randomly generate two classes according to different Gaussian distributions (different means and covariance matrices). Fig.

6(a) shows the computational time with different numbers of samples (from 50 to 1,000). Note that when the number of samples exceeds 200, CVX-SSC halts due to the memory limits in Matlab. Therefore, we only report the computational time when the number of samples is smaller than 200 for CVX-SSC. In contrast, the proposed scalable LD-SSC1 and LD-SSC2 can deal with more than 1,000 samples for cluster-

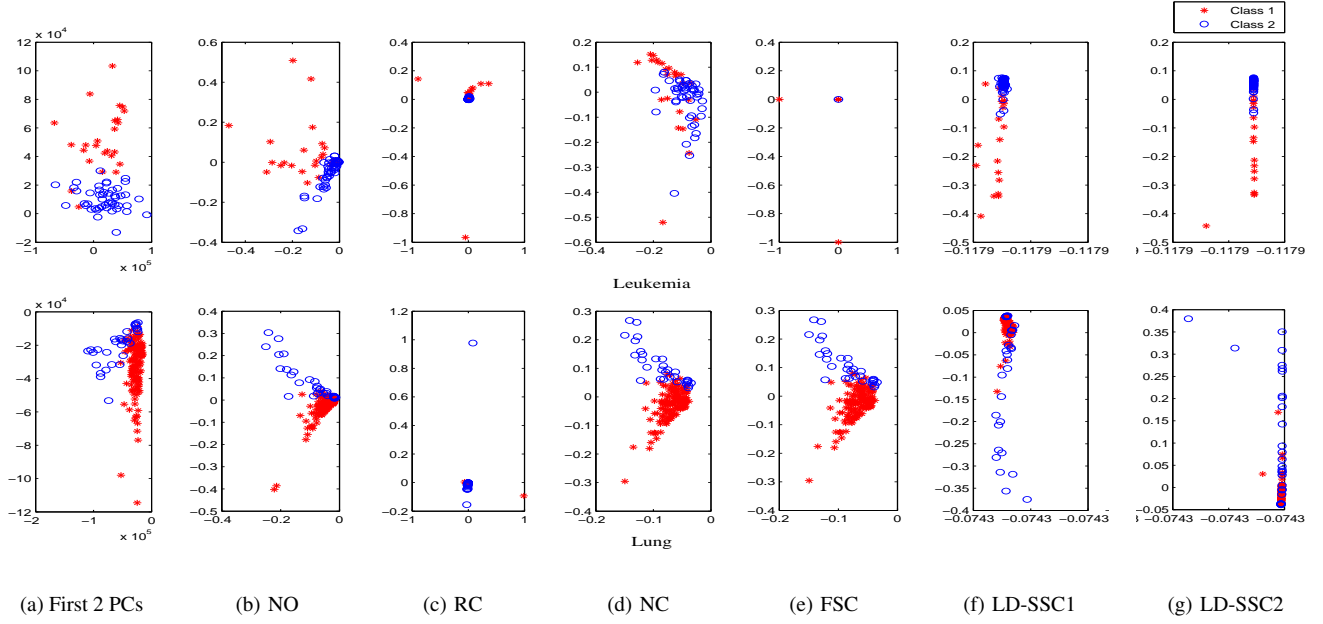


Fig. 4. Embedded results (formed by the two principal eigenvectors of the normalized affinity matrix) obtained by the different clustering algorithms given the fixed kernel parameter ($d = 2$) for the cancer data sets. The first row shows the results from Leukemia and the second row shows the results from Lung.

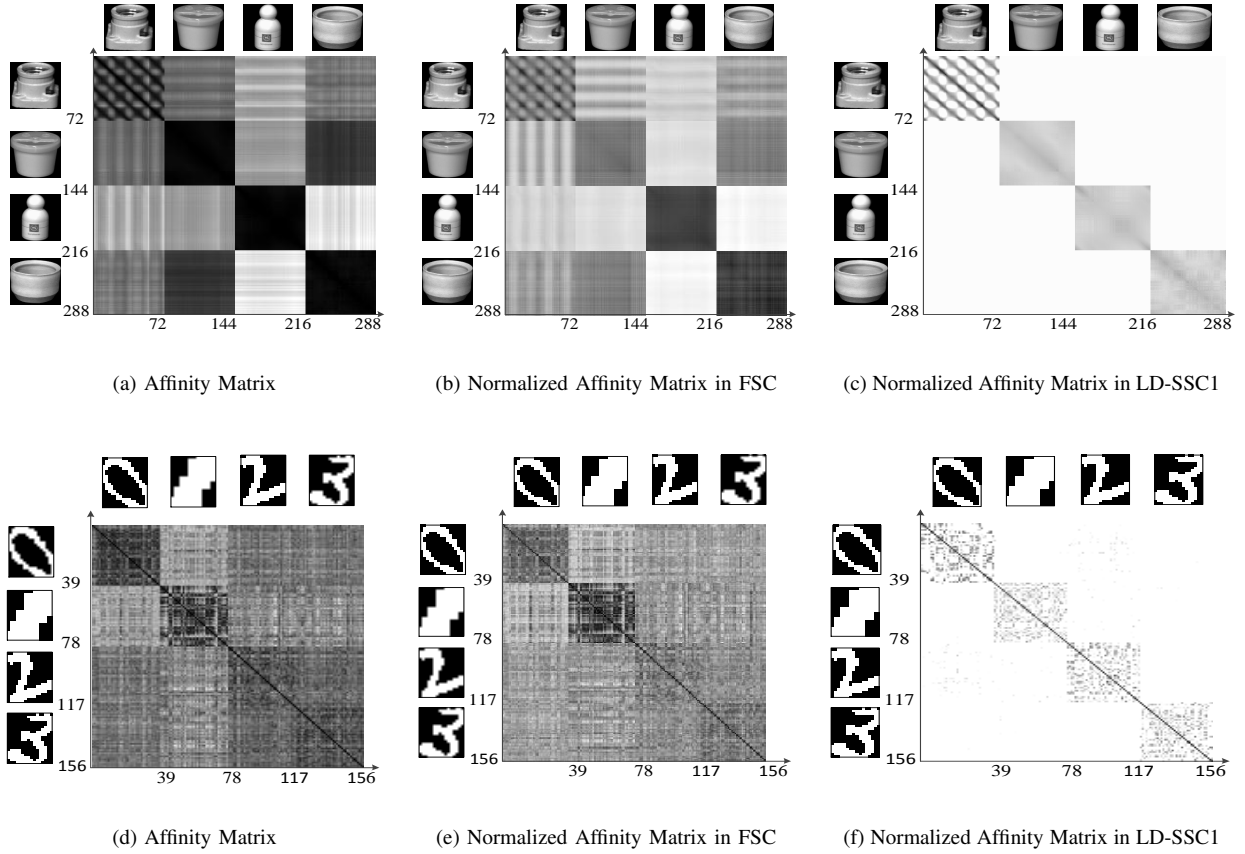


Fig. 5. Comparison of the obtained normalized affinity matrix between FSC and LD-SSC1 (given the fixed kernel parameter). The first row shows some results for COIL-20 (four classes and 72 samples for each class) and the second row shows some results for Alphadigits (four classes and 39 samples for each class). A sample from each class is also given. Darker pixels mean that they are more similar. The overlaps between different classes are reduced significantly after the Frobenius normalization with the p.s.d constraint (see (c) and (f)). Note that LD-SSC2 is not shown since LD-SSC2 obtains a similar normalized affinity matrix as LD-SSC1.

TABLE IV

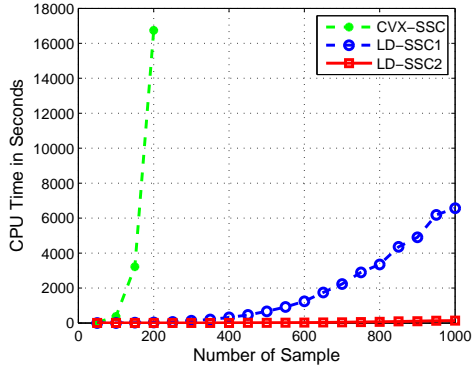
COMPARISON OF THE LOWEST ERROR RATE AND COMPUTATIONAL TIME (SECONDS) BETWEEN THE CVX SOLVER AND THE PROPOSED ALGORITHM. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Lowest Error Rate												
Algorithm	SPECTF	Wine	Pima	Hayes-Roth	Iris	BUPA	Leukemia	Lung	ORL	Yale	COIL-20	Alphadigits
CVX-SSC	—	0.2697	—	0.5857	0.1133	—	0.1806	0.0994	—	0.4909	—	—
LD-SSC1	0.1873	0.2697	0.3411	0.4688	0.0867	0.4203	0.1806	0.0994	0.6400	0.4182	0.2431	0.4950
LD-SSC2	0.1873	0.2697	0.3411	0.4688	0.0867	0.4203	0.1806	0.0994	0.6450	0.4182	0.2431	0.4932
Computational Time												
Algorithm	SPECTF	Wine	Pima	Hayes-Roth	Iris	BUPA	Leukemia	Lung	ORL	Yale	COIL-20	Alphadigits
CVX-SSC	—	6682s	—	3353s	1195s	—	68s	3421s	—	3675s	—	—
LD-SSC1	153s	23s	4344s	45s	13s	298s	4s	31s	448s	18s	5106s	5037s
LD-SSC2	7s	2s	69s	2s	2s	9s	2s	3s	17s	1s	243s	184s

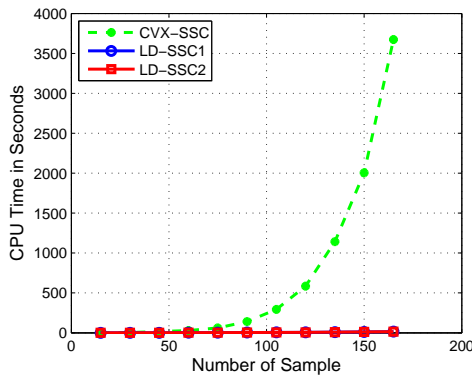
TABLE V

MEMORY USAGE (BYTE) OF ALL THE PARAMETERS FOR L-BFGS-B IN LD-SSC1 AND LD-SSC2.

Memory usage of L-BFGS-B												
Algorithm	SPECTF	Wine	Pima	Hayes-Roth	Iris	BUPA	Leukemia	Lung	ORL	Yale	COIL-20	Alphadigits
LD-SSC1	1.09M	0.49M	9.01M	0.39M	0.35M	1.82M	0.08M	0.50M	2.45M	0.42M	31.7M	30.1M
LD-SSC2	2.73M	1.21M	22.52M	0.98M	0.86M	4.54M	0.20M	1.25M	6.11M	1.04M	79.1M	75.2M



(a) Toy data



(b) Yale

Fig. 6. Comparison of the computational time between LD-SSC1, LD-SSC2 and CVX-SSC.

ing. For the large scale clustering, the general purpose SDP solvers are not viable, but our scalable algorithm is applicable by exploiting the dual form of the SSC problem.

Then, we show the clustering results on a real data set. Fig.

6(b) gives the computational time on the Yale face database with different numbers of samples for clustering, using CVX-SSC and the proposed algorithm. It is obvious to observe that the proposed LD-SSC1 and LD-SSC2 are much faster than CVX-SSC. LD-SSC1 and LD-SSC2, which use the special structure in the dual form, achieve a higher efficiency than the general-purpose SDP solver.

Finally, Table IV gives a comparison of the lowest error rate and the corresponding computational time of CVX-SSC for all the data sets given the fixed kernel parameters. Note that all the computational time of CVX-SSC for SPECTF, Pima, BUPA, ORL, COIL-20, and Alphadigits data sets is not shown, because the number of samples in these data sets exceeds 200, which makes CVX-SSC not applicable. CVX-SSC, LD-SSC1 and LD-SSC2 have similar error rate for most data sets. However, the proposed LD-SSC1 and LD-SSC2 are much more efficient than CVX-SSC. In Table V, we also report the memory usage of all the parameters for L-BFGS-B which consumes the majority of the computational time in LD-SSC1 and LD-SSC2. LD-SSC2 has a faster convergence rate, but requires more memory than LD-SSC1 at each iteration in L-BFGS-B. This is due to the fact that LD-SSC2 finds the solution of \mathbf{Q} and \mathbf{u} , but LD-SSC1 only obtains the solution of \mathbf{u} in L-BFGS-B.

E. Image Segmentation Results

In this subsection, we explore the application of the proposed algorithm and show real image segmentation with our and other competing clustering algorithms. The framework of [31] is used to perform different clustering algorithms for real image segmentation. Images are convolved with oriented filter pairs to extract the magnitude of edge responses. The pixel affinity matrix \mathbf{K} is measured based on the maximum magnitude of edges across the line between two pixels [38]. For convenience, we resize all the images to the size of 40×40 . In all the segmentation experiments, the number of classes k is manually chosen ($k = 4$ in our experiments). Similar to the previous experiments, the only difference between these image segmentation approaches is the normalization algorithm used.

Fig. 7 shows the image segmentation results on a set of face images. We observe that LD-SSC1 and LD-SSC2 outperform NO, NC, RC, and FSC in most cases. Both LD-SSC1 and LD-SSC2 are able to accurately locate the boundary of the object and remove small segmentation patches. FSC, however, has the over-segmentation problem at the area inside the object. In contrast, semidefinite spectral clustering via Lagrange duality yields more accurate segmentation results than the traditional spectral clustering algorithms.



(a) NO (b) RC (c) NC (d) FSC (e) LD-SSC1 (f) LD-SSC2

Fig. 7. Comparisons of the different multiclass segmentation results on real face images.

V. CONCLUSION AND DISCUSSION

Normalization of the affinity matrix is a crucial factor for spectral clustering. Existing normalization algorithms can be considered as the doubly stochastic approximation to the affinity matrix under different error measures. In this paper, an efficient and scalable normalization algorithm with two versions (i.e., LD-SSC1 and LD-SSC2) for semidefinite spectral clustering is presented. The two versions are equivalent for the SSC problem but differ only in their optimization step, where LD-SSC1 requires less memory usage while LD-SSC2 has faster convergence rate. We show that it is more desirable to have the doubly stochastic constraint as well as the p.s.d. constraint during the normalization step.

The proposed LD-SSC1 and LD-SSC2 are simpler and much more scalable than the standard interior-point based SDP solvers. The key to our algorithm is to exploit the Lagrange dual form by using the structure of the optimization problem. Experimental results on various data sets have shown the importance of the normalization and the p.s.d. constraint to the final performance of clustering. The proposed algorithm achieves better performance than the state-of-the-art algorithms in most data sets. We also observe that the L_1 normalization (Ratio-cut) or the Relative entropy normalization (Normalized-cut) can sometimes degrade the clustering performance in some data sets compared with the case without

normalization. On the contrary, the proposed LD-SSC1 and LD-SSC2 improve the clustering performance in most cases.

REFERENCES

- [1] A. K. Jain, M. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.
- [2] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, B. C., Canada, 2001, pp. 849–856.
- [3] A. Pothen, H. D. Simon, and K. P. Liou, "Partitioning sparse matrices with eigenvectors of graph," *SIAM Journal of Matrix Anal. Appl.*, vol. 11, pp. 430–452, 1990.
- [4] L. Hagen and A. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 11, no. 9, pp. 1074–1085, 1992.
- [5] P. K. Chan, M. D.F. Schlag, and J. Y. Zien, "Spectral k-way ratio-cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 13, no. 9, pp. 1088–1096, 1994.
- [6] F. Chung, "Spectral graph theory," AMS publication, 1997.
- [7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.
- [8] C. Ding, X. He, H. Zha, M. Gu, and H. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *Proc. IEEE Int. Conf. Data Mining*, Washington, D.C., USA, 2001, pp. 107–114.
- [9] F. R. Bach and M. I. Jordan, "Learning spectral clustering with application to speech separation," *J. Mach. Learn. Research*, vol. 7, pp. 1963–2001, 2006.
- [10] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, B. C., Canada, 2002, pp. 585–591.
- [11] Y. Yang, D. Xu, F. P. Nie, S. C. Yan, and Y. T. Zhuang, "Image clustering using local discriminant models and global integration," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2761–2772, 2010.
- [12] C. Ding, "A tutorial on spectral clustering," Talk presented at ICML 2004. (slides available at <http://ranger.uta.edu/~chqding/Spectral/>)
- [13] F. Nie, C. Ding, D. Luo, H. Huang, "Improved minmax cut graph clustering with nonnegative relaxation," in *Proc. European Conf. on Mach. Learn. / Principles and Pract. of Know. Discov. in Databases*, Barcelona, Spain, 2010, pp. 451–466.
- [14] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, C. Zhang, "Spectral embedded clustering: a framework for in-sample and out-of-sample spectral clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1796–1808, 2011.
- [15] D. Luo, C. Ding, H. Huang, "Consensus spectral clustering in near-linear time," in *Proc. IEEE Conf. Data Eng.*, Hannover, Germany, pp. 1079–1090, 2011.
- [16] D. Luo, C. Ding, H. Huang, F. Nie, "Forging the graphs: a low rank and positive semidefinite graph learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, Nevada, USA, 2012, pp. 2969–2977.
- [17] W. Y. Chen, Y. Song, H. Bai, C. J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems," *IEEE Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 568–586, 2011.
- [18] R. Zass and A. Shashua, "A unifying approach to hard and probabilistic clustering," in *Proc. IEEE Int. Conf. Comp. Vis.*, Beijing, China, 2005, pp. 294–301.
- [19] R. Zass and A. Shashua, "Doubly stochastic normalization for spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, B. C., Canada, 2006, pp. 1569–1576.
- [20] W. Liu and S. Chang, "Robust multi-class transductive learning with graphs," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, Miami, Florida, USA, 2009, pp. 381–388.
- [21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [22] C. Zhu, R. H. Byrd, and J. Nocedal, "L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization," *ACM Trans. Mathematical Software*, vol. 23, no. 4, pp. 550–560, 1997.
- [23] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means, spectral clustering and normalized cuts," in *Proc. ACM Int. Conf. Knowledge & Discovery Data Mining*, Seattle, USA, 2004, pp. 555–556.
- [24] R. Sinkhorn and P. Knopp, "Concerning non-negative matrices and doubly stochastic matrices," *Pacific J. Math.*, vol. 21, pp. 343–348, 1967.
- [25] D'Esopo, D., "A convex programming procedure," *Naval Logistics Quarterly*, vol. 6, no. 1, pp. 33–42, 1959.

- [26] J. Borwein and A. Lewis, *Convex Analysis and Nonlinear Optimization*, Springer-Verlag, New York, 2000.
- [27] J. Von Neumann, "Functional Operators Vol. II," Princeton University Press, 1950.
- [28] E. P. Xing and M. I. Jordan, "On semidefinite relaxation for normalized k-cut and connections to spectral clustering," Division of Computer Science, University of California, Berkeley, Technical Report CSD-03-1265, 2003.
- [29] K. Q. Weinberger, F. Sha, Q. Zhu, and L. Saul, "Graph Laplacian regularization for large-scale semidefinite programming," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, B. C., Canada, pp. 1489–1496, 2007.
- [30] P. Biswas, K. Toh, and Y. Ye, "A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation," *SIAM Journal on Scien. Comput.*, vol. 30, no. 3, pp. 1251–1277, 2008.
- [31] S. X. Yu and J. Shi, "Multiclass spectral clustering," in *Proc. IEEE Int. Conf. Comp. Vis.*, Pittsburgh, PA, USA, 2003, pp. 313–319.
- [32] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," Columbia Univ., Technical Report CUCS-006-96, Feb. 1996.
- [33] D. Cai, X. F. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, 2005.
- [34] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.
- [35] P. N. Belhumeur, J. P. Heapanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997.
- [36] C. Ding, "Document retrieval and clustering: from principal component analysis to self-aggregation networks," *Proc. Int. Workshop on Artificial Intelligence and Statistics*, Key West, Florida, 2003.
- [37] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming. Version 1.22," <http://cvxr.com/cvx/>
- [38] S. X. Yu, "Computational models of perceptual organization," *Ph.D. thesis*, Carnegie Mellon University, May 2003.